Comparing Proof Techniques on the Simply Typed Lambda Calculus between ITT and HoTT

Yanjun Yang, 2020

Advisor: David Walker

March 4, 2019

1 Extrinsically Typed $\lambda^{\rightarrow}$ with de Brujin Indices in ITT

First, we worked with an extrinsically typed lambda calculus. In an extrinsically typed lambda calculus, the type of the lambda term exists separately from the lambda term itself. In this language, there are two types: a base type Boolean which contains normal forms are True and False, and a type-constructor $t \rightarrow t'$ whose normal forms are all well-typed lambda abstractions. There are additionally two other terms in the language, function application and variables, whose names are de Brujin indices. Contexts in this language are an ordered list of types of terms. The following is the definition in Agda:

```
data Type : Set where
  Boolean : Type
  Function : Type  Type  Type


data Type-Box : Type  Set where
  Box : (t : Type)  Type-Box t
```

```
data Context : Set where
  Empty : Context
  _,_ : Context  Type  Context

Variable : Context  Set
Variable Empty =
Variable ($\Gamma$ , t) = (Variable $\Gamma$)  (Type-Box t)

data Term ($\Gamma$ : Context) : Set where
  Var : Variable $\Gamma$  Term $\Gamma$
  Fun :  t  Term ($\Gamma$ , t)  Term $\Gamma$
  App : Term $\Gamma$  Term $\Gamma$  Term $\Gamma$
  True : Term $\Gamma$
  False : Term $\Gamma$
```

2 Intrinsically Typed $\lambda^{\rightarrow}$ with de Brujin Indices in ITT